




10-1990

Transputers at Work: Real-Time Distributed Robot Control

Louis L. Whitcomb
Yale University

Daniel E. Koditschek
University of Pennsylvania, kod@seas.upenn.edu

Follow this and additional works at: http://repository.upenn.edu/ease_papers

 Part of the [Electrical and Computer Engineering Commons](#), and the [Systems Engineering Commons](#)

Recommended Citation

Louis L. Whitcomb and Daniel E. Koditschek, "Transputers at Work: Real-Time Distributed Robot Control", *Proceedings of North American Transputer Users Group Meeting*, 107-118. October 1990.

Accepted for *Proceedings of the 1990 North American Transputer Users Group Meeting*, October 1990.

NOTE: At the time of publication, author Daniel Koditschek was affiliated with Yale University. Currently, he is a faculty member in the Department of Electrical and Systems Engineering at the University of Pennsylvania.

This paper is posted at ScholarlyCommons. http://repository.upenn.edu/ease_papers/677
For more information, please contact repository@pobox.upenn.edu.

Transputers at Work: Real-Time Distributed Robot Control

Abstract

An advanced robot control system joining a GMF A-500 industrial arm with a network of Inmos Transputers is described in the context of the developing field of robotics. The robot system is used to experimentally compare conventional linear control algorithm performance with both the advanced “computer torque” inverse dynamics control algorithm and a recently developed “adaptive computed torque” algorithm.

Disciplines

Electrical and Computer Engineering | Engineering | Systems Engineering

Comments

Accepted for *Proceedings of the 1990 North American Transputer Users Group Meeting*, October 1990.

NOTE: At the time of publication, author Daniel Koditschek was affiliated with Yale University. Currently, he is a faculty member in the Department of Electrical and Systems Engineering at the University of Pennsylvania.

Transputers at Work: Real-Time Distributed Robot Control.

Louis L. Whitcomb * and Daniel E. Koditschek †
Center for Systems Science
Yale University, Department of Electrical Engineering

Abstract

An advanced robot control system joining a GMF A-500 industrial arm with a network of Inmos Transputers is described in the context of the developing field of robotics. The robot system is used to experimentally compare conventional linear control algorithm performance with both the advanced "computed torque" inverse dynamics control algorithm and a recently developed "adaptive computed torque" algorithm.

Keywords: Adaptive, Control, Real-Time, Robot.

1 Introduction

In this paper we present recent advances in robot arm control at the Yale Robotics Lab, and attempt to portray them in the context of the emerging field of robotics. Section 2 sketches the practice, and inherent defects, of current industrial robot control. Section 3 briefly outlines advances in several constituent technologies which are enabling rapid advances in industrial robot systems. Section 4 presents a state of the art robot system which provides improved trajectory tracking performance by joining the most advanced available motors and microprocessors with recently developed model-based control algorithms. A performance comparison with conventional robot control algorithms is presented.

2 Industrial Robot Control Today

Conventional industrial robot arms are articulated limbs with rotational or sliding (prismatic) joints (each with a motor) with a "hand" or tool for performing a task. Dangerous manufacturing environments were the earliest large scale applications of robots - for example spray painting, where the environment is highly toxic, and machine-tool loading, which can be physically dangerous to the operator. By the mid 1980's, as robot technology matured beyond its infancy, manufacturing practice expanded to include both spot welding and arc welding, materials handling such as stacking, and simple assembly tasks. All of these tasks, however, share the following two common traits: First, they are "slow" enough to work well with simple controllers. Second, the robot can do these tasks "blind" - they require only moderate accuracy and little or no sensing.

The digital controllers currently in use in every major industrial robot are of such restricted computational power that they are unable to compensate for system errors due to

*E.E. Department, 1968 Yale Station, New Haven CT 06520 arpanet: llw@crowin.eng.yale.edu

†This work was supported in part by SGS Thomson-INMOS Corporation, GMF Robotics Corporation, and the National Science Foundation under a Presidential Young Investigator Award held by the second author.

mutual interaction of the robot joints. Traditional robot control algorithms have the disadvantage that they fail to compensate for natural joint interactions such as inertial coupling, coriolis forces, and torque coupling - and allow them to appear as systematic position errors of the robot "hand" as it tracks a desired trajectory. These traditional control algorithms have the advantage, however, that they typically can be executed on a single small and inexpensive microprocessor. Thus robot design engineers have been forced to sacrifice robot performance to achieve reasonable hardware costs.

3 Advanced Robot Control: Driven by Technology

The areas of market growth for industrial robot systems are those which require faster and more nimble performance than current systems can provide. In this section we suggest that the use of advanced control systems is the outgrowth of advances in several disparate fields whose components are used in robot systems. Motor design, composite materials, sensors, and microprocessors represent but a few of the most important robot components. All have rapidly advanced in the last decade.

Motors: A decade ago the hydraulic actuators of early robots gave way to electric servos. Now, new materials (rare-earth magnets) and new designs (permanent magnet [Ele80] and variable-reluctance[Man89, RKBK90] motors) provide electric motors with superior torque-to-weight ratios and bandwidth. Improved speed and accuracy is afforded by these motors in direct-drive robots which obviate the friction, compliance, and backlash inherent in gearbox-drive robots. Advances in FET power transistors have enabled motor designers to replace the inherently unreliable mechanical commutators and analog amplifiers with solid state amplifiers which are maintenance free, deliver greater currents, and are more energy-efficient. Exotic new technologies wherein the distinctions between robot and motor are abandoned are being explored [SH88].

Materials: Composite materials offer impressive strength-to-weight ratios while maintaining structural stiffness. Composites are beginning to replace heavy metal castings and assemblies in some advanced prototypes [KK88].

Sensors: The variety and magnitude of advances in sensing technology are incredible. Indeed, even an outline of this intensely active field would be almost immediately obsolete. The general trend, however, is clear - increasingly fast and accurate sensors of all varieties are becoming smaller and less expensive. The ease and economy with which we can now obtain sensor information is in stark contrast to our primitive ability to process and interpret this information.

Microprocessors: The last decade has seen a several order of magnitude reduction in the cost of embedded microprocessors, digital signal processors, and floating-point computational engines. This has sparked a revolution in the field of control system engineering: where digital control systems have surpassed analog designs as the architecture of choice.

Control Algorithms: Several advanced robot control algorithms which "understand" the whole robot have been proposed to compensate for dynamics such as joint interactions [KKT84, LVS86, KK86, AAH88] - including some algorithms which "learn" about the robot as they work [CHS86, SH87, SL86, OS88, Kod87]. The advantage of the advanced over traditional algorithms is that they promise superior speed and accuracy performance. Their disadvantage is that they are often several orders of magnitude more computationally complex than traditional robot controllers - and may exceed the capability of even our fastest computers. In fact, some of these algorithms are sufficiently complex that they have never actually been used in practice - only simulated!

4 An Advanced Robot Control Architecture

In this section we discuss the design and implementation of the Yale-GMF A-500 Robot. We had set out with the intention of building “merely” an easily re-configurable and incrementally cheap family of computational engines in order to get on with the “real work” of robotics. We quickly discovered that the very ubiquity of design that so facilitated interconnection led to a potentially bewildering variation in network behavior depending upon apparently innocent changes in topology and communication protocols, and is explored in [WK90]. Herein we restrict discussion to basic aspects of the control system.

4.1 T8 Based Computational Hardware

The computational hardware for these implementations is the Yale XP/DCS [F. 89], a powerful real-time control node based on the INMOS Transputer floating point microprocessor. We had set out with the intention of building “merely” an easily re-configurable and incrementally cheap family of computational engines. Students at the Yale robotics Lab currently use the XP/DCS *in actual working systems* which include: A robot juggler [BKK89], a GMF A-500 SCARA arm [BWLK89] a field-rate real-time vision system, an advanced sensor-based obstacle avoidance system [CL89], a variable-reluctance motor commutation test bed, and a one degree of freedom pneumatic-muscle robot. Several new projects are underway.

The excellent scalar floating point performance of the T8 transputer, coupled with their powerful and easily reconfigurable DMA links have provided a computational environment of of superior real-time performance and flexibility.

4.2 Robot Control Network: Control and Actuation

Where does the control system end and the plant begin? While the conceptual distinction between the two entities may be clear, the physical division is ambiguous. Is the division at the output shaft from the electric motor to the link? Is it at the command input to the motor power amplifiers? Or is it at the diodes of an optical shaft encoder?

We have found it convenient to place this conceptual division within the processor network itself, partitioning the network into an **actuator subsystem** considered to be part of the plant and a **control subsystem** which constitutes the controller. Each robot joint has a dedicated **actuator processor** which provides a device independent joint interface to the control subsystem. It handles the multitude of housekeeping task necessary to run an actual motor — possibly including commutation, position and velocity estimation, safety monitoring, and the like. The actuator process is a smart device driver which provides the control subsystem with a uniform interface to the many (often infuriatingly different) motor hardware interfaces of the robot.

Each actuator process has one channel to and one channel from the control subsystem. To the control subsystem it provides current state information — position and velocity of the joint — in floating point format in conventional units. From the control subsystem it receives torque (or force) commands, also in floating point format and conventional units. This modular design limits device dependent design to the actuator processors and frees the control subsystem to perform the purely “algorithmic” task of executing a control law.

We feel that the distinction between actuator and controller subsystems offers implementation advantages, by modularizing device dependence, as well as the conceptual clarity of isolating the more abstract algorithmic and computational control functions within the controller subsystem.

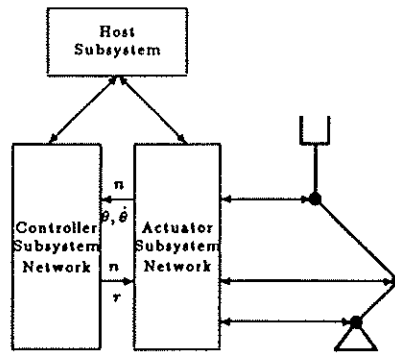


Figure 1: Control System Structure

4.3 The Robot Plant

The GMF Robotics Model A-500, a four degree of freedom SCARA type arm shown in Figure 2, was chosen as the target mechanical unit. Each joint which has a motor capable of delivering torque and a position sensor. Like virtually all commercially available robot systems, the original A-500 system controller provides an integrated high level user interface which serves admirably in industrial applications, but precludes the low level servo intervention which is needed in the research laboratory. It was therefore necessary to replace the manufacturer's control system with our own low level interface. At present interfaces are fully operational for the two primary revolute axes.

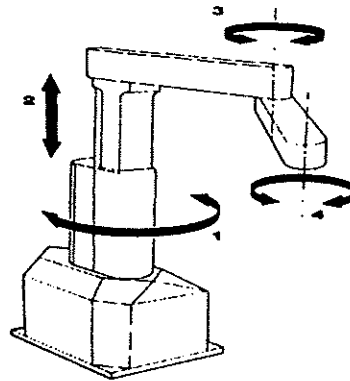


Figure 2: The GMF Model A-500

5 Actual Performance: Two Advanced Robot Control Algorithms

5.1 The Robot Trajectory Tracking Problem

The *robot control problem* is generally stated as follows: A *reference trajectory* is a desired sequence of positions for the robot joints which is parameterized as a function of time. It is the controller's task to command torque at each robot joint so that the robot joint trajectories

exactly follow the reference trajectory. The block diagram structure of the closed loop system comprising robot and controller is shown in figure 3.

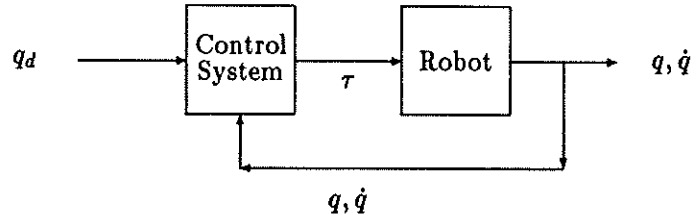


Figure 3: Closed Loop System.

Here q_d, \dot{q}_d is the reference trajectory position and velocity vectors (functions of time), q, \dot{q} are the actual joint position and velocity vectors, and τ is a vector of control torques or forces exerted on each joint by the actuators.

A generally accepted model for robot arm dynamics takes the form ¹

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} = \tau \quad (1)$$

where $M(q)$ arises from the inertial properties of the links, and $C(q, \dot{q})\dot{q}$ from the coriolis and centripetal forces of motion.

5.2 Three Control Algorithms for Robot Tracking

Three control laws which provide solutions to the *robot control problem* are the Proportional-Derivative (PD), Nonlinear Computed Torque (NLCT) and Adaptive Computed Torque (ANLCT) control algorithms.

5.2.1 The Proportional-Derivative Control Law

The PD control law (eq. 2) uses the difference between the desired and actual positions and velocities, amplified by the fixed gain matrices K_p and K_d respectively, to obtain actuator torque commands.

$$\tau = K_d(\dot{q} - \dot{q}_d) + K_p(q - q_d) \quad (2)$$

Because of the computational simplicity obtained when K_p and K_d are diagonal, the PD control law is employed, almost without exception, in all existing industrial robots. This control law, however, ignores the complicated dynamics of the robot arm, and may deliver poor high-speed performance.

¹This equation is easily derived, as in the appendix of [WBK88]. Representative terms of the $C(q, \dot{q})\dot{q}$ matrix for two robots are listed in appendix A.

5.2.2 The Nonlinear Computed Torque Control Law

As of this writing, the most generally discussed control algorithm to achieve robot tracking is the “computed torque control algorithm” [Fre83, LWP80, TBIC84], of which we present one version (eq. 3). The technique proposes to *exactly cancel* the complicated nonlinear terms in 1, and use the traditional engineering technique of “pole placement” on the resulting linear system.

$$\tau = C(q, \dot{q})\dot{q} + M(q)[\ddot{q}_d] + K_d(\dot{q} - \dot{q}_d) + K_p(q - q_d) \quad (3)$$

This algorithm has the advantage that it promises excellent tracking performance - one can show that the tracking errors eventually vanish. There are two disadvantages: First, this algorithm is sufficiently computationally intensive (see appendix) that only recently has it been implemented exactly in real-time [KKT84, LVS86, KK86, AAH88]. Second, the algorithm requires *exact* knowledge of both the robot geometric (spatial relationship between joints) and inertial (moments of inertia) parameters. In practice, robot geometric parameters are usually known accurately. The robot inertial parameters, however, are usually unknown and difficult to obtain [AAH88].

Without exact robot inertial parameters, the advantage promised by this controller is compromised. Even if the inertial parameters of a robot were measured exactly, they are instantly compromised when the robot gripper picks up a payload or tool. The usefulness of this algorithm in actual applications is the continuing subject of empirical investigation.

5.2.3 The Adaptive Nonlinear Computed Torque Control Law

To correct deficiencies of the NLCT control algorithm in the presence of unknown inertial parameters, several researchers [CHS86, SH87, SL86, OS88, Kod87]. have proposed algorithms which “learn” the robot’s inertial parameters. These “adaptive control” algorithms exploit structure of the nonlinear robot equations of motion 1 where the inertial terms enter *linearly* [AAH88]. Thus the nonlinear terms $M(q)$ and $C(q, \dot{q})\dot{q}$ can be factored as a product of the constant matrices \hat{M} and \hat{C} and the time varying matrices $\tilde{M}(q)$ and $\tilde{C}(q, \dot{q})$.

$$M(q) = \hat{M}\tilde{M}(q) \quad (4)$$

$$C(q, \dot{q})\dot{q} = \hat{C}\tilde{C}(q, \dot{q})\dot{q} \quad (5)$$

In [Kod87] the following feedback control law is proposed

$$\tau = \hat{C}\tilde{C}(q, \dot{q})\dot{q} + \hat{M}\tilde{M}(q)[\ddot{q}_d] + K_d(\dot{q} - \dot{q}_d) + K_p(q - q_d) \quad (6)$$

with a parameter update law which “learns” the robot’s inertial parameters:

$$\dot{\hat{M}} = -(\epsilon(q - q_d) + (\dot{q} - \dot{q}_d))[\tilde{M}(q)\ddot{q}_d]^T \quad (7)$$

$$\dot{\hat{C}} = -(\epsilon(q - q_d) + (\dot{q} - \dot{q}_d))[\tilde{C}(q, \dot{q})\dot{q}_d]^T \quad (8)$$

This control algorithm performs feedback control to provide accurate tracking (eq 6) while simultaneously “learning” the robot’s inertial distribution (eq 7). It is shown in [Kod87] that as the adaptive controller’s knowledge increases, the robot tracking performance improves.

5.3 Experimental Comparison of the Three Control Algorithms

In this section we compare the actual performance of the three control algorithms implemented for the two revolute joints of the Yale-GMF A-500 SCARA arm. The control algorithms were implemented exactly (without omitting any terms) using 32-bit floating point representation on a single T8 transputer at fixed rate of 800Hz. As described in the previous section, low-level housekeeping for each of the two joints was performed at each joint by a dedicated transputer, as depicted in figure 1

The first two revolute degrees of freedom were used in these experiments for two reasons: First, the revolute-revolute SCARA configuration represents one of the simplest kinematic chains possessing nonlinear dynamics. Second, the two joints represent the following different drive technologies: The “elbow joint” (link 0), a Fanuc Model 1-0 3 phase DC brushless motor, delivers a maximum torque of 16 N-m to a 47:1 spiroidal gear box. It is representative of the highly geared drive systems used in most conventional industrial robots. The “shoulder joint” (link 1), a 14” NSK direct-drive variable-reluctance (VR) motor capable of delivering a static torque of 250 N-m. The VR motors are a newly developed technology offering superior torque/weight ratios and direct drive capabilities as described in section 3.

5.3.1 Tracking Error

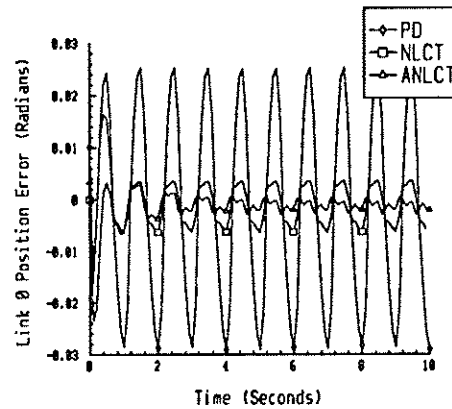


Figure 4: Link 0 Position Error vs. Time

Figure 4 shows the tracking position error of link 0 for the three controllers as a function of time. The robot’s initial position and velocity were zero, and the following reference trajectory was chosen

$$q_{d0}(t) = 0.0 \quad (9)$$

$$q_{d1}(t) = 1.0 \sin(\pi t) \quad (10)$$

to illustrate the dynamic coupling between the two links. Joint 0, beginning with zero state error and tracking a zero reference trajectory, is disturbed by torques generated at joint 1. The feedback gains were identical for each of the three algorithms.

The PD control algorithm, which ignores these nonlinear coupling torques, allows steady state errors of about 0.025 Radian (about 1.5 degree).

The NLCT control algorithm proposes to exactly cancel dynamic coupling terms between joints based on a fixed internal model, and is seen to provide improved tracking. While

theory predicts that the NLCT algorithm should provide asymptotically exact tracking, the actual robot has a steady state tracking error which can be attributed to two major defects: First, the controller requires exact parameter matching between its internal model and the actual robot. Exact parameter measurement - and therefore matching - is often difficult to obtain, as mentioned in section 5.2.2. Parameter matching errors give rise to trajectory tracking error. Second, our theoretical analysis uses rigid-body classical mechanics models of the robot mechanical unit. The rigid body model fails to capture many analytically complicated phenomenon such as structural and drive flexibility, motor electrical dynamics, power electronics dynamics, unmodeled gearbox dynamics such as backlash and hysteresis, and various nonlinear friction effects. All of these effects are present in various degrees, and their modeling and control represent active fields of research.

The ANLCT algorithm provides a solution to the exact parameter matching defects of the NLCT algorithm by using the (easy to measure) kinematic parameters of the robot and adaptively "learning" the (difficult to measure) inertial parameters. Its tracking performance is superior to both previous algorithms. Here again, however, tracking is not perfect and this algorithm has the same unmodeled dynamics defects as the NLCT algorithm.

5.3.2 Tracking Error Norms

We wish to compare controller performance over a variety of reference trajectories - including position (and perhaps velocity) errors as a basis for judgment. While curves in figure 4, representing the position tracking error for a single link as a function of time, provide a palpable representation of tracking performance - the visual comparison of successive such graphs becomes an act of aesthetic judgment rather than empirical analysis.

A scalar valued norm used as an objective measure of tracking performance is the L_2 norm:

$$L_2[e(t)] = \frac{1}{t} \left(\int_{t_0}^t \|e(t)\|^2 dt \right)^{\frac{1}{2}} \quad (11)$$

where $e(t)$ is the scalar (or vector) valued tracking error in question. The norm measures an "average" of the tracking error, thus a smaller L_2 norm indicates better performance. We shall apply this norm to the position errors of both links, $e_0(t)$ and $e_1(t)$, for several reference trajectories.

$$e_0(t) = q_0(t) - q_{d0}(t) \quad (12)$$

$$e_1(t) = q_1(t) - q_{d1}(t) \quad (13)$$

In the following example, the reference trajectories

$$q_{d0}(t) = 0.125 \sin(\alpha \omega t) \quad (14)$$

$$q_{d1}(t) = 0.25 \sin(\omega t) \quad (15)$$

were used. The parameter $\alpha = 0.789$ was arbitrarily selected so that the joints will not be in phase. The parameter ω was set at values of 2, 4, 6, and 8 radians/second for successive runs of 300 seconds (5 minutes) of each control algorithm.

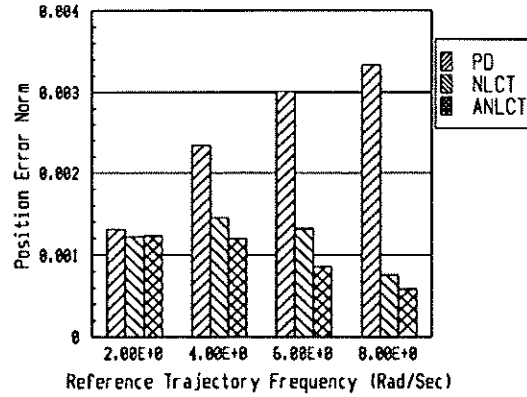


Figure 5: L_2 norm: Link 0 Position Error

Figure 5 shows the L_2 norm of $e_0(t)$ (link zero) for each controller at each of four ω values. This graph agrees with the previous plot - indicating that NLCT controller is uniformly better to the PD controller, and the ANLCT controller is generally superior to both.

It is also clear that these three controllers, which are distinguished at higher frequencies, perform identically for low frequency reference trajectories. We attribute this low frequency similarity to two causes: First, an examination of the control laws will reveal that the non-linear terms vanish for very low velocities and accelerations - leaving identical feedback laws. Second, unmodeled effects of friction, backlash, hysteresis, drive imperfections, and the like may become significant (even dominant) at slow velocities and accelerations.

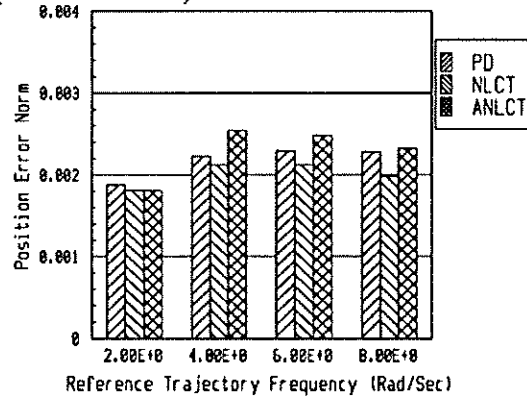


Figure 6: L_2 norm: Link 1 Position Error

Figure 6 shows strikingly different L_2 norm of $e_1(t)$ (link one) for each controller at the four ω values. Here again the NLCT algorithm performs only marginally better than the PD algorithm, and the ANLCT algorithm performs marginally worse! Here we believe the unmodeled dynamics of Link 1's gearbox dominate, causing the three controllers to perform almost uniformly poorly.

6 Conclusion

We have sketched recent advances in robot control techniques, as well as advances in disparate technologies which have enabled these advances. We have described a working distributed real-time robot controller for a GMF A-500 industrial robot arm. This controller utilizes both the elegant networking capability and the intrinsic floating point power of the Inmos

Transputer.

Two advanced robot control algorithms were implemented and experimentally compared with a conventional control algorithm. The computational performance of the controller on the NLCT algorithm compares favorably with other implementations reported in the literature. The ANLCT algorithm was implemented and tested for the first time. Actual performance comparisons, figure 5, demonstrated that the advanced control algorithms offer significant performance advantages. It is also shown, figure 6, that advanced control algorithms may perform no better than conventional techniques when controlling characterized and inaccurately modeled mechanical systems. These preliminary results suggest that advanced control techniques offer significant tracking performance improvements when judiciously applied to mechanical systems, with the caveat that even a "good" controller cannot always compensate for a "bad" mechanical system.

We are currently implementing the aforementioned control algorithms for a recently completed three degree of freedom direct-drive robot arm. The clean mechanical design and VR motor technology employed in this machine promise torque and speed capabilities, which absolutely require advanced control techniques, promise to enable such dexterous dynamic tasks as three dimensional robot juggling, catching, throwing and, perhaps, a less than respectable game of squash.

A Equations of Motion

It is well known that the computational complexity of the robot equations of motion, depending upon the formulation chosen, grows at least linearly with the number of joints. As these equations are an intrinsic part of model-based controllers, it is incumbent on us to evaluate them in real time. As an example to suggest that this is a non-trivial task, here we list representative elements of the equations of motion for both a three and six degree of freedom robot.

The following is the explicit symbolic derivation of the [1,2] element of the $C(q, \dot{q})\dot{q}$ coriolis matrix for a simple three degree of freedom planar arm:

$$c_{1,2} = -2l_1l_2m_2\dot{\theta}_1 \sin \theta_2 - l_1l_2m_2\dot{\theta}_2 \sin \theta_2 - 2l_1l_2m_3\dot{\theta}_1 \sin \theta_2 - l_1l_2m_3\dot{\theta}_2 \sin \theta_2 - 2l_1l_3m_3\dot{\theta}_1 \sin(\theta_2 + \theta_3) - l_1l_3m_3\dot{\theta}_2 \sin(\theta_2 + \theta_3) - l_1l_3m_3\dot{\theta}_3 \sin(\theta_2 + \theta_3)$$

The following is the exact expression for the [1,2] term of the $C(q, \dot{q})\dot{q}$ coriolis matrix for the Unimation PUMA 560, a six degree of freedom industrial robot arm:

$$\begin{aligned} & \dot{\theta}_1(-1.03937 \cos \theta_3 \sin^2 \theta_2 + 0.0215632 \cos \theta_3 \sin 2\theta_2 + 0.0209956 \cos 2\theta_3 \sin^2 \theta_2 + 0.31519 \cos 2\theta_3 \sin 2\theta_2 + \\ & 1.03937 \cos^2 \theta_2 \cos \theta_3 - 0.0209956 \cos^2 \theta_2 \cos 2\theta_3 + 0.0215632 \cos^2 \theta_2 \sin \theta_3 - 0.0215632 \sin^2 \theta_2 \sin \theta_3 - \\ & 1.03937 \sin 2\theta_2 \sin \theta_3 - 0.00248717 \cos \theta_3 \cos \theta_5 \sin^2 \theta_2 - 0.63038 \cos \theta_3 \sin^2 \theta_2 \sin \theta_3 + 0.0419912 \cos \theta_3 \sin 2\theta_2 \sin \theta_3 + \\ & 2.30400 * 10^{-5} \cos 2\theta_3 \cos 2\theta_4 \sin 2\theta_2 + 1.16928 * 10^{-4} \cos 2\theta_3 \cos \theta_5 \sin^2 \theta_2 + 0.00249466 \cos 2\theta_3 \cos \theta_5 \sin 2\theta_2 + \\ & 6.91200 * 10^{-5} \cos 2\theta_3 \cos 2\theta_5 \sin 2\theta_2 - 0.00248717 \cos \theta_5 \sin 2\theta_2 \sin \theta_3 + 0.00248717 \cos^2 \theta_2 \cos \theta_3 \cos \theta_5 + \\ & 0.63038 \cos^2 \theta_2 \cos \theta_3 \sin \theta_3 - 1.16928 * 10^{-4} \cos^2 \theta_2 \cos 2\theta_3 \cos \theta_5 - 0.00248717 \cos \theta_3 \cos \theta_4 \sin 2\theta_2 \sin \theta_5 + \\ & 4.60800 * 10^{-5} \cos \theta_3 \cos 2\theta_4 \sin^2 \theta_2 \sin \theta_3 - 0.00498931 \cos \theta_3 \cos \theta_5 \sin^2 \theta_2 \sin \theta_3 + 2.33856 * 10^{-4} \cos \theta_3 \cos \theta_5 \sin 2\theta_2 \sin \theta_3 - \\ & 1.38240 * 10^{-4} \cos \theta_3 \cos 2\theta_5 \sin^2 \theta_2 \sin \theta_3 - 0.00249466 \cos 2\theta_3 \cos \theta_4 \sin^2 \theta_2 \sin \theta_5 + 1.16928 * 10^{-4} \cos 2\theta_3 \cos \theta_4 \sin 2\theta_2 \sin \theta_5 + \\ & 2.30400 * 10^{-5} \cos 2\theta_3 \cos 2\theta_4 \cos 2\theta_5 \sin 2\theta_2 + 0.00248717 \cos \theta_4 \sin^2 \theta_2 \sin \theta_3 \sin \theta_5 - 4.60800 * 10^{-5} \cos^2 \theta_2 \cos \theta_3 \cos 2\theta_4 \sin \theta_3 + \\ & 0.00498931 \cos^2 \theta_2 \cos \theta_3 \cos \theta_5 \sin \theta_3 + 1.38240 * 10^{-4} \cos^2 \theta_2 \cos \theta_3 \cos 2\theta_5 \sin \theta_3 + 0.00249466 \cos^2 \theta_2 \cos 2\theta_3 \cos \theta_4 \sin \theta_5 - \\ & 0.00248717 \cos^2 \theta_2 \cos \theta_4 \sin \theta_3 \sin \theta_5 - 2.33856 * 10^{-4} \cos \theta_3 \cos \theta_4 \sin^2 \theta_2 \sin \theta_3 \sin \theta_5 - 0.00498931 \cos \theta_3 \cos \theta_4 \sin 2\theta_2 \sin \theta_3 \sin \\ & 4.60800 * 10^{-5} \cos \theta_3 \cos 2\theta_4 \cos 2\theta_5 \sin^2 \theta_2 \sin \theta_3 - 1.84320 * 10^{-4} \cos 2\theta_3 \cos \theta_4 \cos \theta_5 \sin^2 \theta_2 \sin \theta_5 + 2.33856 * 10^{-4} \\ & 4 \cos^2 \theta_2 \cos \theta_3 \cos \theta_4 \sin \theta_3 \sin \theta_5 + 4.60800 * 10^{-5} \cos^2 \theta_2 \cos \theta_3 \cos 2\theta_4 \cos 2\theta_5 \sin \theta_3 + 1.84320 * 10^{-4} \cos^2 \theta_2 \cos 2\theta_3 \cos \theta_4 \cos \theta_5 \sin \\ & 3.68640 * 10^{-4} \cos \theta_3 \cos \theta_4 \cos \theta_5 \sin 2\theta_2 \sin \theta_3 \sin \theta_5 - 1.03904 \sin 2\theta_2) + \dot{\theta}_2(0.41984 \cos \theta_2 - 0.00374785 \cos \theta_2 \cos \theta_3 + \\ & 0.19026 \cos \theta_2 \sin \theta_3 + 0.19026 \cos \theta_3 \sin \theta_2 + 0.00374785 \sin \theta_2 \sin \theta_3 + 4.32288 * 10^{-4} \cos \theta_2 \cos \theta_5 \sin \theta_3 + \\ & 0.00124358 \cos \theta_2 \sin \theta_4 \sin \theta_5 + 4.32288 * 10^{-4} \cos \theta_3 \cos \theta_5 \sin \theta_2 + 4.60800 * 10^{-5} \cos \theta_2 \cos \theta_3 \cos \theta_4 \sin \theta_4 + \\ & 4.32288 * 10^{-4} \cos \theta_2 \cos \theta_3 \cos \theta_4 \sin \theta_5 - 5.84640 * 10^{-5} \cos \theta_2 \cos \theta_3 \sin \theta_4 \sin \theta_5 + 0.00124733 \cos \theta_2 \sin \theta_3 \sin \theta_4 \sin \theta_5 + \\ & 0.00124733 \cos \theta_3 \sin \theta_2 \sin \theta_4 \sin \theta_5 - 4.60800 * 10^{-5} \cos \theta_4 \sin \theta_2 \sin \theta_3 \sin \theta_4 - 4.32288 * 10^{-4} \cos \theta_4 \sin \theta_2 \sin \theta_3 \sin \theta_5 + \\ & 5.84640 * 10^{-5} \sin \theta_2 \sin \theta_3 \sin \theta_4 \sin \theta_5 - 4.60800 * 10^{-5} \cos \theta_2 \cos \theta_3 \cos \theta_4 \cos 2\theta_5 \sin \theta_4 + 9.21600 * 10^{-5} \cos \theta_2 \cos \theta_5 \sin \theta_3 \sin \theta_4 \sin \end{aligned}$$

$$\begin{aligned}
& 9.21600 \times 10^{-5} \cos \theta_3 \cos \theta_5 \sin \theta_2 \sin \theta_4 \sin \theta_5 + 4.60800 \times 10^{-5} \cos \theta_4 \cos 2\theta_5 \sin \theta_2 \sin \theta_3 \sin \theta_4 + 0.313635 \sin \theta_2) + \\
& \theta_3(-0.00374785 \cos \theta_2 \cos \theta_3 + 0.19026 \cos \theta_2 \sin \theta_3 + 0.19026 \cos \theta_3 \sin \theta_2 + 0.00374785 \sin \theta_2 \sin \theta_3 + \\
& 4.32288 \times 10^{-4} \cos \theta_2 \cos \theta_5 \sin \theta_3 + 4.32288 \times 10^{-4} \cos \theta_3 \cos \theta_5 \sin \theta_2 + 4.60800 \times 10^{-5} \cos \theta_2 \cos \theta_3 \cos \theta_4 \sin \theta_4 + \\
& 4.32288 \times 10^{-4} \cos \theta_2 \cos \theta_3 \cos \theta_4 \sin \theta_5 - 5.84640 \times 10^{-5} \cos \theta_2 \cos \theta_3 \sin \theta_4 \sin \theta_5 + 0.00124733 \cos \theta_2 \sin \theta_3 \sin \theta_4 \sin \theta_5 + \\
& 0.00124733 \cos \theta_3 \sin \theta_2 \sin \theta_4 \sin \theta_5 - 4.60800 \times 10^{-5} \cos \theta_4 \sin \theta_2 \sin \theta_3 \sin \theta_4 - 4.32288 \times 10^{-4} \cos \theta_4 \sin \theta_2 \sin \theta_3 \sin \theta_5 + \\
& 5.84640 \times 10^{-5} \sin \theta_2 \sin \theta_3 \sin \theta_4 \sin \theta_5 - 4.60800 \times 10^{-5} \cos \theta_2 \cos \theta_3 \cos \theta_4 \cos 2\theta_5 \sin \theta_4 + 9.21600 \times 10^{-5} \cos \theta_2 \cos \theta_5 \sin \theta_3 \sin \theta_4 \sin \theta_5 \\
& 9.21600 \times 10^{-5} \cos \theta_3 \cos \theta_5 \sin \theta_2 \sin \theta_4 \sin \theta_5 + 4.60800 \times 10^{-5} \cos \theta_4 \cos 2\theta_5 \sin \theta_2 \sin \theta_3 \sin \theta_4) + \theta_4(-4.60800 \times 10^{-5} \\
& 5 \cos \theta_2 \sin \theta_3 - 4.60800 \times 10^{-5} \cos \theta_3 \sin \theta_2 + 4.60800 \times 10^{-5} \cos \theta_2 \cos 2\theta_5 \sin \theta_3 + 4.60800 \times 10^{-5} \cos \theta_3 \cos 2\theta_5 \sin \theta_2 - \\
& 0.00124358 \cos \theta_4 \sin \theta_2 \sin \theta_5 + 0.00124733 \cos \theta_2 \cos \theta_3 \cos \theta_4 \sin \theta_5 + 5.84640 \times 10^{-5} \cos \theta_2 \cos \theta_4 \sin \theta_3 \sin \theta_5 - \\
& 4.32288 \times 10^{-4} \cos \theta_2 \sin \theta_3 \sin \theta_4 \sin \theta_5 + 5.84640 \times 10^{-5} \cos \theta_3 \cos \theta_4 \sin \theta_2 \sin \theta_5 - 4.32288 \times 10^{-4} \cos \theta_3 \sin \theta_2 \sin \theta_4 \sin \theta_5 - \\
& 0.00124733 \cos \theta_4 \sin \theta_2 \sin \theta_3 \sin \theta_5 + 9.21600 \times 10^{-5} \cos \theta_2 \cos \theta_3 \cos \theta_4 \cos \theta_5 \sin \theta_5 - 9.21600 \times 10^{-5} \cos \theta_4 \cos \theta_5 \sin \theta_2 \sin \theta_3 \sin \theta_5 \\
& \theta_5(9.21600 \times 10^{-5} \cos \theta_2 \cos \theta_3 \sin \theta_4 + 4.32288 \times 10^{-4} \cos \theta_2 \cos \theta_3 \sin \theta_5 - 0.00124358 \cos \theta_5 \sin \theta_2 \sin \theta_4 - \\
& 9.21600 \times 10^{-5} \sin \theta_2 \sin \theta_3 \sin \theta_4 - 4.32288 \times 10^{-4} \sin \theta_2 \sin \theta_3 \sin \theta_5 + 0.00124733 \cos \theta_2 \cos \theta_3 \cos \theta_5 \sin \theta_4 + \\
& 4.32288 \times 10^{-4} \cos \theta_2 \cos \theta_4 \cos \theta_5 \sin \theta_3 + 5.84640 \times 10^{-5} \cos \theta_2 \cos \theta_5 \sin \theta_3 \sin \theta_4 + 4.32288 \times 10^{-4} \cos \theta_3 \cos \theta_4 \cos \theta_5 \sin \theta_2 + \\
& 5.84640 \times 10^{-5} \cos \theta_3 \cos \theta_5 \sin \theta_2 \sin \theta_4 - 0.00124733 \cos \theta_5 \sin \theta_2 \sin \theta_3 \sin \theta_4)
\end{aligned}$$

References

- [AAH88] Chae H. An, Christopher G. Atkeson, and John M. Hollerbach. *Model-Based Control of a Robot Manipulator*. MIT Press, Cambridge, MA, USA, 1988.
- [BKK89] M. Bühler, D. E. Koditschek, and P.J. Kindlmann. A Simple Juggling Robot: Theory and Experimentation. In V. Hayward and O. Khatib, editors, *International Symposium on Experimental Robotics*. Springer-Verlag, 1989.
- [BWLK89] M. Bühler, L. Whitcomb, F. Levin, and D. E. Koditschek. A distributed message passing computational and i/o engine for real-time motion control. In *Proc. American Control Conference*, pages 478-488, Pittsburgh, PA, Jun 1989. American Control Society.
- [CHS86] John J. Craig, Ping Hsu, and S. Sastry. Adaptive control of mechanical manipulators. Memorandum M86/3, Electronics Research Laboratory, University of California, Berkeley, Berkeley, CA, Jan 1986.
- [CL89] Edward Cheung and Vladimir Lumelsky. Development of sensitive skin for a 3d robot arm operating in an uncertain environment. In *IEEE International Conference on Robotics and Automation*, pages 1056-1061, Scottsdale, AZ, USA, 1989.
- [Ele80] Electro-Craft Corporation. *DC Motors, Speed Controls, Servo Systems*. Electro-Craft, Hopkins, MINN, 1980.
- [F. 89] F. Levin and M. Bühler and L. Whitcomb and D. E. Koditschek. Transputer computer juggles real-time robotics. *Electronic Systems Design*, 19(2):77-82, Feb 1989.
- [Fre83] E. Freund. Fast nonlinear control with arbitrary pole placement for industrial robots and manipulators. *The International Journal of Robotics Research*, 1(1):65-78, 1983.
- [KK86] Pradeep K. Khosla and Takeo Kanade. Real-time implementation and evaluation of model-based controls on cmu dd arm ii. In *Proceeding IEEE International Conference on Robotics and Automation*, pages 1546-1555, San Francisco, CA, Apr 1986.
- [KK88] H. Kazerooni and S. Kim. A new architecture for direct drive robots. In *IEEE International Conference on Robotics and Automation*, pages 442-445, Philadelphia, Pennsylvania, USA, 1988.
- [KKT84] Takeo Kanade, Pradeep K. Khosla, and Nobuhiko Tanaka. Real-time control of cmu direct-drive arm ii using customized inverse dynamics. In *IEEE Conference on Decision and Control*, pages 1345-1352, Las Vegas, Nevada, USA, 1984.
- [Kod87] Daniel E. Koditschek. Adaptive techniques for mechanical systems. In *Fifth Yale Workshop on Applications of Adaptive Systems Theory*, pages 259-265, New Haven, CT, May 1987. Center for Systems Science, Yale University.

- [LVS86] M. B. Leahy, Jr., K. P. Valavanis, and G. N. Saridis. The effects of dynamic models on robot control. In *IEEE International Conference on Robotics and Automation*, page 4954, San Francisco, CA, USA, 1986.
- [LWP80] J. Y. S. Luh, M. W. Walker, and R. P. Paul. Resolved acceleration control of mechanical manipulators. *IEEE Transaction on Automatic Control*, AC-25:468-474, 1980.
- [Man89] Manzer Et Al. Variable reluctance motor characterization. *IEEE Transactions on Industrial Electronics*, 36(1), 1989.
- [OS88] R. Ortega and M. W. Spong. Adaptive motion control of rigid robots: A tutorial. In *IEEE Conference on Decision and Control*, pages 1575-1584, Austin, Texas, USA, 1988.
- [RKBK90] A. A. Rizzi, D. E. Koditschek, M. Buehler, and P. J. Kindlmann. Preliminary experiments toward a torque controlled variable reluctance motor. Technical Report 8907, Yale University, Center for Systems Science, New Haven, Connecticut, USA, March 1990.
- [SH87] Nader Sadegh and Roberto Horowitz. Stability analysis of an adaptive controller for robotic manipulators. In *Proceedings IEEE International Conference on Robotics and Automation*, Raleigh, NC, Apr 1987.
- [SH88] S. Salcudean and R. L. Hollis. A magnetically levitated fine motion wrist: Kinematics, dynamics, and control. In *IEEE International Conference on Robotics and Automation*, pages 261-266, Philadelphia, Pennsylvania, USA, 1988.
- [SL86] Jean-Jacques E. Slotine and Weiping Li. On the adaptive control of robot manipulators. In *Proc. ASME Winter Annual Meeting*, Anaheim, CA., Dec 1986.
- [TBIC84] T. J. Tarn, A. K. Bejczy, A. Isidori, and Y. Chen. Nonlinear feedback in robot arm control. In *Proc. 23rd IEEE Conference on Decision and Control*, pages 736-751, Las Vegas, Nev., Dec 1984.
- [WBK88] Louis L. Whitcomb, M. Buehler, and D. E. Koditschek. Preliminary experiments real-time distributed motion control. In *Proc. North American Transputer Users Group*, NY, Oct 1988.
- [WK90] Louis L. Whitcomb and Daniel E. Koditschek. Robot control in a message passing environment: Theoretical questions and preliminary experiments. In *IEEE International Conference on Robotics and Automation*, pages 1198-1203, Cincinnati, Ohio, USA, 1990.